Data Mining CSE 572- Assignment 3 / Mini Project 1 Submission Report

Ankit Nadig (1211213650), anadig@asu.edu
Vraj Delhivala (1211213637), vdelhiva@asu.edu

Problem Statement- Studying the application of k-nearest neighbor(KNN), Neural Network and Support Vector Machine classifiers on two-real world classification problems.

Problem 1

First, we import the data from the respective files into matrices- trainingData (X_train), trainingClass (y_train), testData (X_test), testClass (y_test).

a. We use a KNN classifier to classify data.into respective classes.
   i. We use the 'fitcknn' function from MATLAB to create a Model using Nearest Neighbors =5 , Distance measure= Euclidean, trainingData and trainingClass as parameters.
   ii. Then, the 'predict' function is used on testData which gives us an output of the labels for the classes.
   iii. The label matrix is then compared to the testClass to find Accuracy
   iv. Accuracy Obtained = 90.0238%

b. We use SVM classifiers to classify and predict data into respective classes.
   i. We use the 'fitcsvm' function to train different SVM for every unique class present in the dataset using kernel function as polynomial and order as 2.
   ii. The SVM Models are structs that are stored in an array.
   iii. Then, the 'predict' function is used on the testData and every SVM Model and a score matrix is computed.
   iv. The SVM for a particular class that gives the highest prediction score, it's class is chosen as the class to associate with the input. Using this we form a label matrix.
   v. The label matrix is then compared to the testClass to find Accuracy
   vi. Accuracy Obtained = 96.8442%

Model	**Accuracy**
KNN	90.0238%
SVM	96.8442%

Accuracy Table for Problem1

Problem 2

First, we load the .mat files into the workspace as X_train,y_train,X_test,y_test

a. We use a KNN classifier to classify data.into respective classes.
   i. We use the 'fitcknn' function from MATLAB to create a Model using Nearest Neighbors =5 , Distance measure= Euclidean, X_train and y_train as parameters.
   ii. Then, the 'predict' function is used on X_test which gives us an output of the labels for the classes.
   iii. The label matrix is then compared to the y_test to find Accuracy
   iv. Accuracy Obtained = 98.6000%

b. We use SVM classifiers to classify and predict data into respective classes.
   i. We use the 'fitcsvm' function to train different SVM for every unique class present in the dataset using kernel function as polynomial and order as 2.
   ii. The SVM Models are structs that are stored in an array.
   iii. Then, the 'predict' function is used on the X_test and every SVM Model and a score matrix is computed.
   iv. The SVM for a class that gives the highest prediction score, it's class is chosen as the class to associate with the input. Using this we form a label matrix. The label matrix is then compared to the y_test to find Accuracy.
   v. Accuracy Obtained = 99.5000%

c. We train an Artificial Neural Network to predict the output.
   i. First, we create a 25 x 3500 label matrix, which is a matrix of 0s and 1s with each column having a 1 corresponding to the index of the class.
   ii. Next, we create a Neural net, with one hidden layer and 25 neurons.
   iii. We then train the network using the 'train' function, which takes about 40 mins.
   iv. Next we check the output by passing X_test to the net and getting the index of the class using the 'vec2ind' function.
   v. This is compared with y_test to find the accuracy.
   vi. Accuracy Obtained = 97.00 - 97.4 % as random selection of weight varies.

Model	**Accuracy**
KNN	98.6000%
ANN	97.00 - 97.4%
SVM	99.5000%

Accuracy Table for Problem2